Robust Path Planning for Slope Traversing Under Uncertainty in Slip Prediction

Hiroaki Inotsume, Takashi Kubota, and David Wettergreen

***Abstract*—This letter addresses the path planning problem for a rover on deformable, sloped terrains. As such terrains induce high slip and possible immobilization of rovers, finding a path that avoids critical slip is important for traversing the terrains. However, it is difficult to predict rover slippage precisely, especially on steep slopes, and a certain level of prediction uncertainty is inevitable. Although several path planning algorithms that consider rover slippage have been proposed thus far, they do not consider the uncertainty in slip prediction. This letter proposes a robust path planning algorithm that finds a safe and efficient path based on a chance-constrained planning approach. The proposed algorithm probabilistically guarantees safety against immobilization, with the safety level specified by user-definable parameters. The simulation results demonstrate the effectiveness and flexibility of the proposed algorithm.**

***Index Terms*—Motion and path planning, probability and statistical methods, space robotics and automation, field robotics.**

I. Introduction

In A robotic surface exploration mission of another astral body, such as the Moon or Mars, typically at the design stage of an exploration rover, its mobility system is carefully designed in terms of the payload size and weight limit, so that the vehicle can access some mission-specific target locations. During the mission operation, the rover is typically operated on terrains within the safety margin of its mobility capability, and risky terrains outside of that margin are avoided [1].

However, during the mission, operators and scientists may sometimes find new, scientifically interesting locations that are located over challenging, risky terrains beyond the mobility safety margin. In such situations, the operators are required to decide whether to continue the rover's exploration within the safe terrains, forgoing the interesting locations, or send the vehicle on an exciting, challenging, and possibly riskier ventures, such as traversing steep slopes of sand dunes, ascending towards crusts of mountains, and jumping into craters, finally returning safely with interesting findings. There might be no gentle route, and the vehicle might need to take some risks of mobility hazard or immobilization. Nevertheless, the operators might wish to thoroughly utilize the vehicle's capability and find better, safer, and more efficient routes to overcome terrains and achieve further success on the mission, with a certain measure of safety.

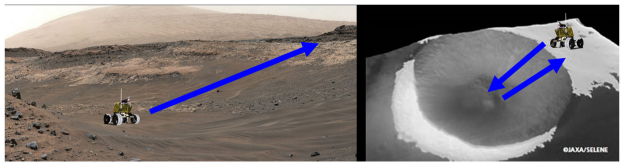


Fig. 1. Sample problems addressed in this study. The scope of this study is to find successful and efficient routes over steep terrains of loose soil. Image credits: NASA/JPL and JAXA.

This letter addresses the issues in path planning to find successful and efficient routes for a rover to traverse slopes of deformable, slip-inducing sand (Fig. 1). Primarily, this study targets terrains with the approximate traversability limit for a rover. A main challenge while navigating a rover on such terrains is that the traversability, including the slip and power requirement, highly depends on multiple factors, such as terrain geometry (inclination and local roughness), terrain types (rocks, bedrock, soil/sand, or their mixture), and the surface conditions (e.g., level of compaction and depth of accumulation) [2], [3]. Hence, it is challenging to predict the traversability precisely.

In the previous study [4], the influence of the slope-ascent direction on the mobility performance of rovers was analyzed. The results indicated that the possibility of successful slope ascent can be improved by selecting an ascent direction that maximizes the energy efficiency under some slip threshold. The present study is based on these findings and applies the key idea to global motion planning over sloped terrains. Specifically, this letter proposes a robust path planning algorithm that finds a safe and efficient path. It is developed based on the chance-constrained planning approaches [15]–[17] to consider slip uncertainty. The effectiveness and flexibility of the proposed algorithm are demonstrated through numerical simulations.

The rest of the paper is organized as follows. Section II reviews the related studies and describes the contributions of this letter. In Section III, the path planning problem is defined first, followed by the details of the proposed algorithm. The effectiveness and flexibility of the proposed algorithm are assessed with sets of numerical simulations in Section IV. Finally, Section V concludes this letter with certain possible directions of future research.

II. Related Work

Numerous studies have been devoted to the navigation and path planning of robotic ground vehicles on natural or rough

Manuscript received September 10, 2019; accepted February 1, 2020. Date of publication February 21, 2020; date of current version March 9, 2020. This letter was recommended for publication by Associate Editor M. Moll and Editor N. Amato upon evaluation of the reviewers' comments. *(Corresponding author: Hiroaki Inotsume.)*

Hiroaki Inotsume is with the Central Research Laboratories, NEC Corporation, Kawasaki, Kanagawa 211-8666, Japan (e-mail: h-inotsume@nec.com).

Takashi Kubota is with the Institute of Space and Astronautical Science, Japan Aerospace Exploration Agency, Sagamihara, Kanagawa 252-5210, Japan (e-mail: kubota@isas.jaxa.jp).

David Wettergreen is with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15217 USA (e-mail: dsw@ri.cmu.edu).

Digital Object Identifier 10.1109/LRA.2020.2975756

terrains (e.g., [5], [6]). These navigation algorithms are sufficiently robust for most outdoor situations, enabling vehicles to navigate toward a goal by autonomously avoiding geometric obstacles, such as large rocks, outcrops, dips, or high slopes. However, particular concerns need to be addressed when navigating vehicles on deformable, slip-inducing terrains, such as sandy slopes, where vehicles can be immobilized or embedded in the soil. These *nongeometric mobility hazards* are challenging to predict because of the complicated vehicle–terrain interaction mechanisms [21], [22].

There is a limited number of studies on motion planning of robotic vehicles for traversing deformable terrains that induce non-negligible slippage. For instance, Helmick *et al.* developed a terrain-adaptive navigation algorithm [7], which predicts rover slippage from terrain appearance and pre-trained regression models. Then, the algorithm generates a feasible path that avoids high slip terrains and hazardous obstacles. Howard and Kelly [8] proposed a trajectory generation algorithm for traversing rough terrain. The algorithm includes vehicle dynamics and vehicle–terrain interaction models, and efficiently generates a smooth, dynamically feasible trajectory. Ishigami *et al.* proposed a multi-path planning algorithm [9] in which multiple candidate paths are generated with varied weight combinations for the cost function. Then, each generated path is evaluated through dynamic simulations with a slip-compensated path-following algorithm. The most feasible path is then automatically selected based on an evaluation index. These methods are promising to a certain extent and have been examined with terrains that include inclines in some areas. However, none of them address problems to overcome steep, challenging slopes where a rover can be potentially immobilized in the soil. Moreover, the models mentioned above do not consider uncertainties in the slip models.

Path planning problems under uncertainties in vehicular motions (and sensory information) have been widely studied for decades [10]–[12]. In addition, those for rough outdoor terrains have also been studied. In [13], uncertainty in robot mobility was considered for rough terrain path planning with an efficient uncertainty propagation method. In [14], the uncertainty in vehicle–terrain interaction was treated by a particle filter-based approach, where multiple state particles are generated during a search by dynamic vehicle simulations with randomly sampled friction coefficients. While these approaches consider mobility uncertainties, the safety of resultant paths is not guaranteed.

Chance-constrained planning approaches [15]–[17] are robust motion planning methods that determine a probabilistically feasible motion plan with the upper-bound of failures. The safety of the resultant paths is probabilistically guaranteed by searching a path under probabilistic constraints, called chance constraints, against some risks. CC-RRT* [16] and RRBT [17] combine the chance-constrained approach with the power of an efficient path search by RRT* algorithm [18]. The chance-constrained path planning algorithms proposed thus far were mainly designed to find a collision-free path under motion and detecting uncertainties. In [19], CC-RRT* was applied to a mobile robot on a deformable terrain where uncertainty models of vehicle slippage were incorporated. However, its main scope was related to collision avoidance, and vehicle entrapment was not explicitly considered.

The contribution of this research is the development of a robust and flexible path planning algorithm that can find safe paths under uncertainty in slip prediction. The present study adopted the chance-constrained planning approach to guarantee safety probabilistically. However, the proposed algorithm does not aim at searching for a path that avoids obstacles, as indicated in the literature. Instead, the main scope of our algorithm is to avoid vehicle immobilization on deformable, sandy terrains. To this end, a slip-oriented chance constraint is developed. Furthermore, this study introduces two user-definable parameters that specify the safety level. This enables the proposed planner to generate paths adaptably depending on the specified safety level.

III. Proposed Algorithm

A. Problem Definition

A path $\mathbf{P} = \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_N\}$ is defined as a sequence of N vehicle states $\mathbf{x}_i$ from the start state $\mathbf{x}_0 = \mathbf{x}_{start}$ to the goal state $\mathbf{x}_N \in \mathbf{X}_{goal}$, where $\mathbf{X}_{goal}$ is either a specific target state or a set of allowable target states.

In chance-constrained path planning, a planner searches for an optimal or sub-optimal path with respect to some specified cost function while probabilistically ensuring that risky states are not adopted. A general form of this chance-constrained path planning problem can be expressed as follows [15]–[17]:

$$\min \ \mathrm{Cost}(\mathbf{P}) = \min \sum_{i=0}^{N-1} \mathbb{E}[c(\mathbf{x}_i, \mathbf{x}_{i+1})] \tag{1}$$

$$\text{s.t.} \quad \mathbb{P}(\bigwedge_i \mathbf{x}_i \notin \mathbf{X}_{risk}) > \delta_{safe} \tag{2}$$

where $c(\mathbf{x}_i, \mathbf{x}_j)$ represents the cost of moving from the state $\mathbf{x}_i$ to $\mathbf{x}_j$. The inequality Eq.(2) represents the chance constraint where $\bigwedge$ represents a conjunction of each constraint $\mathbf{x}_i \notin \mathbf{X}_{risk}$ being satisfied throughout the path from the start to the goal. δ_{safe} denotes the confidence level of the safety against the risk. In general, $\mathbf{X}_{risk}$ can take any form, combination, and set of risky states to be considered.

In this research, the risk of immobilization in the sand is considered one of the most critical risks, and thus, it is specified in a chance constraint. As the immobilization risk increases with slippage in the longitudinal (x) direction of the rover motion, the level of the slip is utilized as an index of such risk. More specifically, the slip $s_x(\mathbf{x}_i, \mathbf{x}_j)$ induced on a path segment from a state x_i to another state x_j is checked if it does not exceed a threshold $s_{x,thre}$:

$$s_x(\mathbf{x}_i, \mathbf{x}_j) < s_{x,thre} \tag{3}$$

Eq. (3) provides a constraint in a deterministic manner. As slip prediction can have a certain level of uncertainty, this research imposes the following chance constraint rather than the deterministic one:

$$\mathbb{P}(\bigwedge_i s_x(\mathbf{x}_i, \mathbf{x}_{i+1}) < s_{x,thre}) > \delta_{safe} \tag{4}$$

Eq.(4) guarantees that the generated path will not induce the slip more than $s_{x,thre}$ over the entire path with the probabilistic confidence of at least δ_{safe}. In the constraint, $s_{x,thre}$ and δ_{safe} are the two parameters that users can specify depending on how much safety to guarantee, or equivalently, how much risk to tolerate.

As a cost function in Eq.(1), the expected energy required for following the path is used in this study to generate an efficient path toward the goal. The energy-based cost function $c(\mathbf{x}_i, \mathbf{x}_j)$

from the state $\mathbf{x}_i$ to $\mathbf{x}_j$ is given as

$$c(\mathbf{x}_i, \mathbf{x}_j) = p(\mathbf{x}_i, \mathbf{x}_j) \cdot t(\mathbf{x}_i, \mathbf{x}_j) \tag{5}$$

where $p(\mathbf{x}_i, \mathbf{x}_j)$ and $t(\mathbf{x}_i, \mathbf{x}_j)$ are the required power and time for the traverse between the two states. Note that both the required power and time depend on vehicle slippage. Therefore, the cost function is also effective in reducing the integral of the slip over the path, or the cumulative immobilization risk.

B. Path Planning Algorithm

In this section, a chance-constrained path planning algorithm is proposed for finding safe and efficient paths over slip-inducing sloped terrains. The ideas of the existing chance-constrained variants of RRT* algorithm [16], [17] are applied and extended for planning paths under prediction uncertainty in the rover slip.

Algorithm 1 shows the proposed path planning algorithm. Note that in the actual algorithm, the search tree comprises state vertices and edges, but the procedures for managing the edges are omitted in Algorithm 1 for brevity. Here, the state vertex $\mathbf{x}$ has the information regarding the corresponding rover position and orientation (x, y, z, θ_y), the control input $\mathbf{x}.u$, the parent state $\mathbf{x}$.parent, the cost to reach the state from the start through the parent $\mathbf{x}$.cost, and the joint probability of the slip not exceeding the threshold during the traverse on the path from the start $\mathbf{x}$.prob.

The overall flow of the algorithm is similar to that of the basic RRT* algorithm [18]. First, the search tree is initialized with the start state (line 1). In each iteration, a new point (x, y) is randomly sampled from the search space, and a new state $\mathbf{x}_{rand}$ is generated (line 3). The height z is estimated from the position and a terrain map. Then, the nearest state $\mathbf{x}_{nearest}$ on the search tree from $\mathbf{x}_{rand}$ is identified (lines 4–11). The feasibility of the motion from $\mathbf{x}_{nearest}$ to $\mathbf{x}_{rand}$ is then checked, and if feasible, $\mathbf{x}_{rand}$ ($\mathbf{x}_{new}$) is connected to one of the tree vertices (lines 12–22). The function Near($\mathbf{T}, \mathbf{x}$) in line 14 finds the tree vertices within some distance from $\mathbf{x}$ as in the original RRT*. After that, rewiring is performed to improve the optimality of the search (lines 23–29) similar to RRT*. The iteration is repeated until either the specified maximum number of iterations M is completed, or the tree reaches the set of the goal states. The proposed algorithm has some modifications from the basic RRT*, such that the chance constraint Eq.(4) is incorporated, and that the possibility of finding a feasible path, if any, is improved. These modifications are described in the following subsections.

1) Chance Constraint Check: The chance constraint violation check is performed by ProbFeasible($\mathbf{x}_i, \mathbf{x}_j$) function (lines 8, 17, and 25). It estimates the probability of slip between the two states $\mathbf{x}_i$ and $\mathbf{x}_j$ not exceeding the slip threshold and then calculates the joint probability throughout the path from the start to x_j. The function returns *true* if the chance constraint Eq.(4) is satisfied, i.e., the joint probability is higher than δ_{safe}, or *false* otherwise.

2) Local Steering: The Steer($\mathbf{x}_i, \mathbf{x}_j$) function (lines 7, 16, and 24) generates a temporary state $\mathbf{x}_{tmp}$ on the line segment from $\mathbf{x}_i$ to $\mathbf{x}_j$ at distance Δl away from $\mathbf{x}_i$.

Steer($\mathbf{x}_i, \mathbf{x}_j$) also computes the local steering input that compensates for the predicted path deviation, or lateral slippage s_y, during the traverse from $\mathbf{x}_i$ to $\mathbf{x}_j$ to make the vehicle reach the target state $\mathbf{x}_j$. The input can either be a single command or a set of sequential commands. In this study, the counter-steering angle is determined as the angle in the inverse direction of the lateral slip with its magnitude the same as the slip angle corresponding to the predictive mean of the lateral slip. The steering input also determines the orientation (yaw) of the rover at the terminal state $\mathbf{x}_j$.

Algorithm 1: Proposed Path Planning Algorithm.

1: $\mathbf{T} \leftarrow \{\mathbf{x}_{start}\}$; $i \leftarrow 0$;
2: **while** $i < M$ **or** goal state is not reached **do**
3: $\mathbf{x}_{rand} \leftarrow$ Sample()
4: $\mathbf{X}_{nearests} \leftarrow$ Nearest($\mathbf{T}, \mathbf{x}_{rand}, k$)
5: $\mathbf{x}_{new} \leftarrow Null$
6: **for** all $\mathbf{x}_j$ in $\mathbf{X}_{nearests}$ **do**
7: $\mathbf{x}_{tmp} \leftarrow$ Steer($\mathbf{x}_j, \mathbf{x}_{rand}$)
8: **if** ProbFeasible($\mathbf{x}_j, \mathbf{x}_{tmp}$) $\wedge$ $\mathbf{x}_j$.cost+$c(\mathbf{x}_j, \mathbf{x}_{tmp}) < \mathbf{x}_{new}$.cost **then**
9: $\mathbf{x}_{new} \leftarrow \mathbf{x}_{tmp}$; $\mathbf{x}_{nearest} \leftarrow \mathbf{x}_j$;
10: **end if**
11: **end for**
12: **if** Probabilistically feasible ($\mathbf{x}_{nearest}, \mathbf{x}_{new}$) exists **then**
13: $\mathbf{x}_{min} \leftarrow \mathbf{x}_{nearest}$
14: $\mathbf{X}_{near} \leftarrow$ Near($\mathbf{T}, \mathbf{x}_{new}$)
15: **for** all $\mathbf{x}_{near}$ in $\mathbf{X}_{near}$ **do**
16: $\mathbf{x}_{tmp} \leftarrow$ Steer($\mathbf{x}_{near}, \mathbf{x}_{new}$)
17: **if** ProbFeasible($\mathbf{x}_{near}, \mathbf{x}_{tmp}$) $\wedge$ $\mathbf{x}_{near}$.cost+$c(\mathbf{x}_{near}, \mathbf{x}_{tmp}) < \mathbf{x}_{new}$.cost **then**
18: $\mathbf{x}_{new} \leftarrow \mathbf{x}_{tmp}$; $\mathbf{x}_{min} \leftarrow \mathbf{x}_{near}$;
19: **end if**
20: **end for**
21: $\mathbf{x}_{new}$.parent $\leftarrow \mathbf{x}_{min}$
22: $\mathbf{T} \leftarrow \mathbf{T} \cup \mathbf{x}_{new}$
23: **for** $\mathbf{x}_{near}$ in $\mathbf{X}_{near} \backslash \mathbf{x}_{min}$ **do**
24: $\mathbf{x}_{tmp} \leftarrow$ Steer($\mathbf{x}_{new}, \mathbf{x}_{near}$)
25: **if** ProbFeasible($\mathbf{x}_{new}, \mathbf{x}_{tmp}$) $\wedge$ $\mathbf{x}_{new}$.cost + $c(\mathbf{x}_{new}, \mathbf{x}_{tmp}) < \mathbf{x}_{near}$.cost **then**
26: $\mathbf{x}_{near} \leftarrow \mathbf{x}_{tmp}$; $\mathbf{x}_{near}$.parent $\leftarrow \mathbf{x}_{new}$;
27: $\mathbf{T} \leftarrow$ Propagate($\mathbf{T}, \mathbf{x}_{near}$)
28: **end if**
29: **end for**
30: **end if**
31: $i \leftarrow i + 1$
32: **end while**

On a sloped terrain, especially steep slopes, an aggressive turning action can induce a large downhill slide or lateral slippage. Therefore, another constraint related to the change in the rover yaw is incorporated in this research. If the difference in the yaw θ_y between $\mathbf{x}_i$ and $\mathbf{x}_j$ is greater than the threshold θ_{thre}, then $\mathbf{x}_j$ is discarded.

Meanwhile, Steer($\mathbf{x}_i, \mathbf{x}_j$) predicts the rover slip and power consumption during the traversal between the two states, and then, the traversal cost is calculated. How to predict the slip and power is described in Section III-C.

3) k-Nearest Vertex Selection: When connecting the sampled state $\mathbf{x}_{rand}$ to the tree $\mathbf{T}$, instead of identifying the single nearest state from $\mathbf{x}_{rand}$, as in the original RRT*, the proposed algorithm searches for the k-nearest states $\mathbf{X}_{nearests}$ on the search tree. Then, for each state in $\mathbf{X}_{nearests}$, the probabilistic feasibility and cost of the traverse from that state to $\mathbf{x}_{rand}$ ($\mathbf{x}_{new}$) are calculated, and a feasible set of $\mathbf{x}_{new}$ and $\mathbf{x}_{nearest}$ with the minimum cost is chosen to step forward (lines 4–11). Similar ideas can be found in [13], [20].

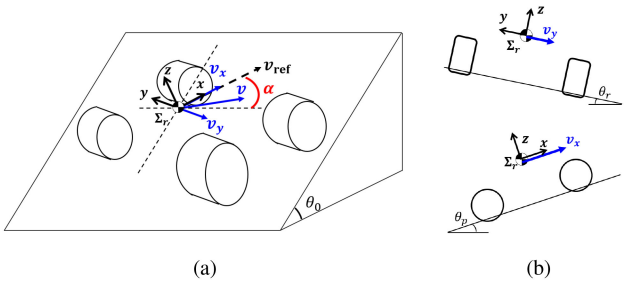


Fig. 2. Schematic view of a rover that is ascending a slope of θ_0 at an angle of attack α.

The main motivation for this modification is that on steeper terrains, the rover slip increases, and the possibility of connecting the single nearest state to a sampled point reduces. In such situations, the newly sampled point can be discarded, even though the second- or third-nearest states can be connected to it. The above modification can reduce the rate of discarding newly sampled states and improve the possibility of finding a path at the cost of additional computation.

4) Rewiring: In the rewiring procedure (lines 23–29), if $\mathbf{x}_{near}$ is rewired to $\mathbf{x}_{new}$, the cost and probability of the constraint satisfaction of $\mathbf{x}_{near}$ are updated along with its parent state. Then, these updates are propagated to the descendants of $\mathbf{x}_{near}$. These calculations are performed by the Propagate($\mathbf{T}, \mathbf{x}_{near}$) function in line 27.

C. Terrain Traversability Prediction

The proposed algorithm requires the prediction of terrain traversability, i.e., slippage and power consumption of a rover, from one state to another.

While traveling on a natural terrain, a rover can experience slippage in the longitudinal and lateral directions [21] (Fig. 2). The slip in rover orientation can also occur while turning or traveling on slopes. However, this study assumes that the rate of yaw slip is trivial and can be canceled out by appropriate orientation control. The longitudinal slip of the vehicle is measured by the slip ratio s_x:

$$s_x = \begin{cases} 1 - v_x/v_{ref} \ (\text{if} \quad v_x \leq v_{ref}) \\ v_{ref}/v_x - 1 \ (\text{if} \quad v_x > v_{ref}) \end{cases} \tag{6}$$

where v_{ref} is the commanded reference velocity and v_x is the actual longitudinal velocity. s_x varies from -1 to 1, with $s_x = 1$ representing a complete slip state where the rover cannot make any forward progress. The negative slip indicates that the rover moves faster than the reference velocity. On the other hand, the lateral slippage can be expressed using the lateral slip ratio, s_y, in a similar fashion to the longitudinal slip ratio as

$$s_y = v_y/v_{ref} \tag{7}$$

where v_y denotes the lateral slip velocity of the rover.

The extent of the slip, along with the power consumption p, depends on terrain geometry and surface composition or condition. Therefore, it is reasonable to model these traversability quantities as functions of the terrain geometry and surface features. However, usually, it is difficult to model these functions without any uncertainty. In this study, the uncertainties in s_x, s_y, and p are assumed to follow zero-mean Gaussian distributions, with their variance depending on terrain geometry $g = (\theta_p, \theta_r)$, where θ_p and θ_r are the pitch and roll angles of the slope, respectively. This study further assumes that the uncertainties in s_x, s_y, and p are independently and identically distributed given the terrain geometry g. That is,

$$s_x = \hat{s}_x(g) + \epsilon_x(g), \quad \epsilon_x(g) \sim \mathbf{N}(0, \sigma_x^2(g)) \tag{8}$$

$$s_y = \hat{s}_y(g) + \epsilon_y(g), \quad \epsilon_y(g) \sim \mathbf{N}(0, \sigma_y^2(g)) \tag{9}$$

$$p = \hat{p}(g) + \epsilon_p(g), \quad \epsilon_p(g) \sim \mathbf{N}(0, \sigma_p^2(g)) \tag{10}$$

where $\hat{\bullet}$ represents the expected mean, $\epsilon_\bullet$ represents the uncertainty, and $\sigma_\bullet^2$ represents the variance. Note that this study assumes the above iid assumption on each slip event for simplifying the path computation. On natural terrains, slippage at one location is highly likely correlated to that of near locations. Therefore, it may be more accurate to model the slip probability as a function of spatial information in addition to terrain geometry as in [24]. That model improvement and evaluations of the impact of that modification on generated paths will be a future work of this research.

The present study assumes that the geometric map of the target environment is available from orbital observations and onboard sensors of the rover and that the soil properties or traversability model of the environment are obtained from some prior traversal data of similar environments. Under these assumptions, the slip and power consumption of moving from the state $\mathbf{x}_i$ to $\mathbf{x}_j$ are predicted in the following procedures. First, a local best-fit plane is calculated around the state $\mathbf{x}_i$ based on the height information of the surrounding area. Then, the pitch and roll angles of the slope or rover are estimated based on the inclination of the best-fit plane and the heading of the rover (Fig. 2). Note that as a path following control is introduced to compensate for the predicted path deviation, the rover heading is not necessarily coincident with the direct direction between the two states. Finally, the mean and variance of the rover slippage in its longitudinal and lateral directions and power consumption, are predicted for the given pitch & roll pair based on the traversability models. The slip and power consumption can be predicted based on either dynamic simulations with some treatments of parameter uncertainties (e.g., [23]) or learning-based regression approaches (e.g., [24]).

IV. Path Planning Simulation

Two sets of numerical simulations were conducted to evaluate the effectiveness of the proposed algorithm and to analyze the influence of the user-specified safety parameters on the generated path qualities.

In the first simulation set, the effectiveness of the proposed algorithm with slip uncertainty consideration was evaluated against a deterministic approach. The synthesized crater-type terrain shown on top of Fig. 3 was used. The terrain was mainly threefold: the crater bottom, inner rim, and outside surface. The bottom part of the crater was a surface of approximately 0°. The inner rim comprised of steep slopes with the inclination ranging from approximately 15° to 28°. The outside of the crater was an almost flat terrain. The local surface profile was generated based on a fractal generation method. In the second simulation set, the influences of the two user-specified parameters, s_{thre} and δ, in the chance constraint Eq.(4) were analyzed. For this purpose, the simple plateau-type terrain depicted at the bottom of Fig. 3 was used. The inclination of the slope surface was set to 25°. The surface of both terrains was covered with lunar regolith.

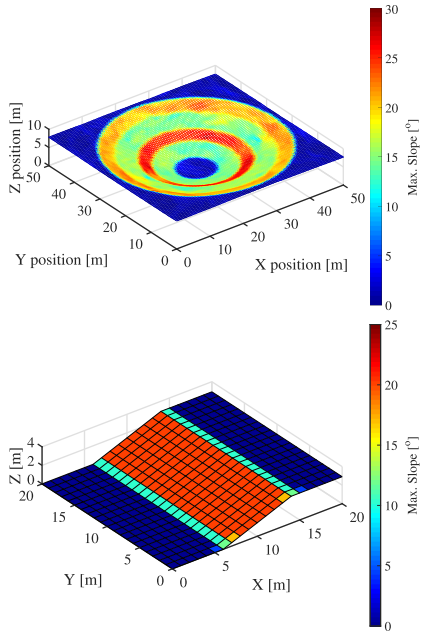


Fig. 3. Synthetic terrains for path planning simulations. The color indicates the maximum inclination angle at each mesh. High slopes are shown in red, whereas the flat terrain is shown in blue. Top: Crater terrain. The slope angle of the inner rim varies from approximately 15° to 28°. Bottom: Plateau terrain for parameter studies. The inclination of the slope is 25°.

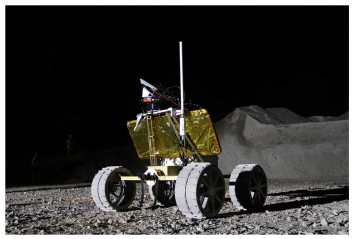


Fig. 4. Rover for the path planning simulations.

The model of the lunar rover prototype (Fig. 4) developed at Carnegie Mellon was used in the simulations. The rover is relatively light-weight and small-sized, with a mass of approximately 25 kg and dimensions of approximately 0.9 m length and 1.1 m width. It has four independently actuated driving wheels with a diameter of 0.32 m.

The terrain traversability models for the rover on lunar regolith, i.e., the relationships between s_x, s_y, and p and the terrain geometry (pitch and roll angles), were derived based on terramechanic-based vehicle–terrain interaction simulations. A detailed description of the models can be found in [4]. As it is computationally expensive to run a physical simulation every time a new state is sampled, this research utilized pre-trained regression models instead, to speed up the path search. Specifically, s_x, s_y, and p were pre-computed for some combinations of the sloping pitch and roll angles from the simulations, and then, a regression model for each was trained based on Gaussian process regression [25] prior to the planning. The derived regression models were used as the predictive mean for the given terrain geometry. A similar approach can be found in [26], where the power consumption model of a rover was trained from a set of dynamic simulation data by using a neural network.

As the uncertainty models for s_x, s_y, and p, artificial uncertainties that follow the Gaussian distributions were introduced to evaluate the proposed algorithm. Fig. 5 shows the traversability models with uncertainties used in this simulation study. In this study, the uncertainties were assumed to increase on steeper slopes. The reasons behind this assumption are that on steeper terrain, the influence of local terrain roughness on rover slippage increases and that the traverse data from which models are trained are typically sparse on risky terrains than benign terrains in actual missions.

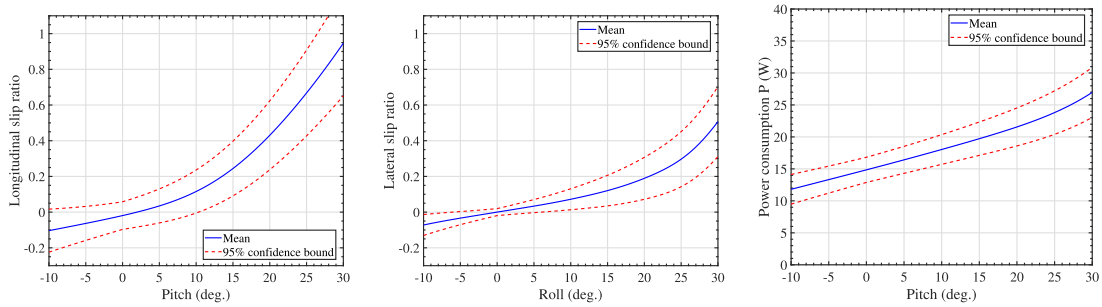
The proposed path planning algorithm was implemented in MATLAB, and run on a computer with a 3.2 GHz processor and 16 GB RAM. In both simulation sets, 10-nearest neighbor vertices were searched for connecting a newly sampled state to the tree. The maximum length of each tree edge was limited to $\Delta l = 1$ m, and the threshold in an orientation change between any two states was set to $\theta_{thre} = 30°$. The maximum iteration of the search was set to 20000, and the algorithm was terminated once the goal was reached.

A. Simulation 1: Comparison of Paths w/ and w/o the Slip Uncertainty Consideration

1) Setting and Procedures: In this simulation set, the effectiveness of the proposed algorithm was evaluated by comparing the paths generated with and without the consideration of model uncertainties. In path planning with uncertainty consideration (proposed method), the violation check of the chance constraint Eq.(4) was performed during the search. However, during planning without uncertainty consideration, the deterministic slip check Eq.(3) was employed, which utilizes only the mean slip model. The slip threshold and probability bound were set as $s_{x,thre} = 0.8$ and $\delta_{safe} = 0.95$, respectively, for comparing the paths generated based on the two approaches with the same slip threshold. In addition to this, the proposed robust planning approach was compared to the deterministic approach with a more conservative, lower slip threshold ($s_{x,thre} = 0.5$). The latter corresponds to the state of the practice approach, where a conservative threshold is used for implicitly addressing prediction uncertainties or errors.

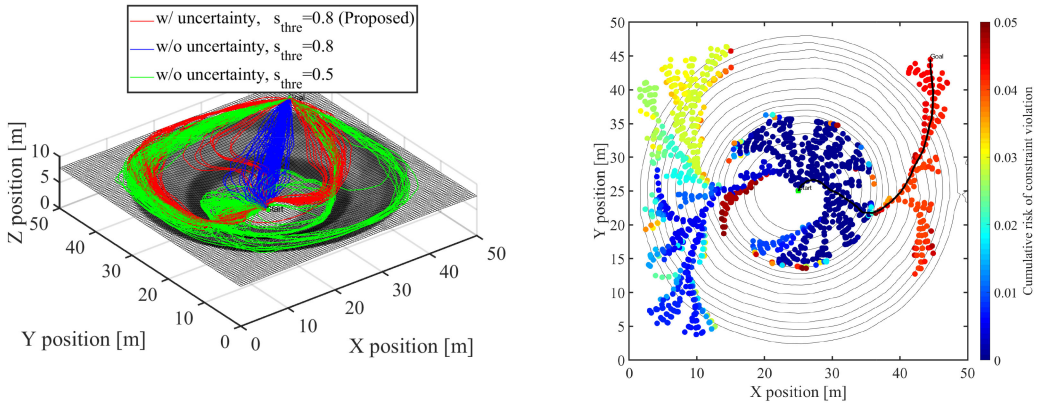
The start location of the rover was set at around the center of the crater bottom, ($(x, y) = (25, 25)$), and the goal was set to outside of the crater, (45, 45). The rover heading at the start state was set to the North (positive y-axis) direction. The heading at the goal position was not specified.

As the proposed algorithm is a random sampling-based planner, 100 paths were generated for each condition to evaluate the planners statistically. Then, 500 Monte Carlo simulations of executing each generated path were performed to evaluate the qualities of the paths under model uncertainties. In the Monte Carlo simulations, the uncertainties ϵ in s_x, s_y, and p along the path were sampled from the Gaussian distributions. The maximum longitudinal slippage and energy consumption during the path execution were estimated for each simulation. Then, the mean of these values from the 500 simulations was calculated. Moreover, this study calculated two types of success rates of each path. Success rate 1 is defined as the ratio of the number of path executions in which the slip s_x never exceeds the threshold to the total number of path executions (i.e., 500). Success rate 2 is, on the other hand, represents the number of path executions that can successfully reach the goal without immobilization. Finally, the



(a) s_x vs θ_p at $\theta_r = 0$ (b) s_y vs θ_r at $\theta_p = 0$ (c) p vs θ_p at $\theta_r = 0$

Fig. 5. Traversability prediction models with uncertainties. The predictive means are plotted with blue curves. The red dashed curves represent 95% confidence bounds of the models. Note that the mean longitudinal slip is slightly lower than zero at $\theta_p = 0^\circ$ (the rover drives slightly faster than commanded) due to the wheel surface profiles.



(a) 100 paths w/ and w/o uncertainty consideration (b) Search tree with risk propagation

Fig. 6. Paths generated in simulation 1. (a) The generated 100 paths with and without slip uncertainty consideration. The terrain color represents the inclination, with darker indicating higher slopes. With $s_{x,thre} = 0.8$, the paths without considering uncertainty (blue) head directly toward the goal, while those with uncertainty (red) take detoured, diagonal slope-ascent routes. (b) A sample search tree with uncertainty consideration. The vertices are colored by the accumulated risk of constraint violation.

mean and standard deviations of these quantities were estimated for the 100 generated paths.

2) Results: Fig. 6(a) shows the 100 paths generated with and without the uncertainty considered. The paths generated without slip uncertainty (blue) tended to head toward the goal directly. However, the paths that considered slip uncertainty (red) detoured, taking diagonal slope-ascent routes.

A sample search tree of a path generated with uncertainty consideration is shown in Fig. 6(b). In this figure, the vertices of the constructed search tree are plotted, with the color representing the accumulated risk of slip constraint violation. The hotter color indicates a higher risk. The contour lines of the crater are depicted in the background. As seen in the figure, the risk of constraint violation grows on the steep slope of the inner rim. The sampled state vertices on the direct line from the start to the goal were rejected during the search because the risk of constraint violation exceeded the threshold bound. Instead, the tree grew toward the East and West sides of the rim, where the inclinations are slightly less steep than those in the Northern and Southern slopes. The west slope is locally less steep than the Eastern one; therefore, the risk tends to be less accumulated on the Western slope. In this example, the tree reached the goal through the East slope.

The statistics of the 100 path computations and the Monte Carlo simulations of path execution are listed in Table I. The energy consumption of the paths without uncertainty consideration was lower than that with uncertainty consideration, as the former took shorter routes. However, the maximum slip experienced along the former paths was higher than the latter. Fig. 7 shows sample slip profiles along one of the 100 paths generated with and without slip uncertainty consideration. In this figure, 500 Monte Carlo simulation results are plotted. Without considering the slip uncertainty, the rover is highly likely to experience a slip higher than the threshold of $s_{x,thre} = 0.8$ around the travel distance of 11–14 m. Furthermore, the slip ratio approaches 1.0, meaning that there is a possibility of immobilization along the path. The mean success rate 1 and 2 of the paths without slip uncertainty are 0.132 and 0.902, respectively. With the slip uncertainty considered, however, the rover slip is less likely to go beyond the threshold, as seen in Fig. 7. Indeed, the mean success rate 1 and 2 of the generated paths by the proposed algorithm are 0.976 and 1.000, respectively. The former value corresponds to the selected probability bound of the chance constraint $\delta_{safe} = 0.95$.

When a lower slip threshold, $s_{x,thre} = 0.5$, was imposed on the deterministic approach, the paths that took longer routes

TABLE I
STATISTICS OF PATH PLANNING SIMULATION 1; 500 MONTE CARLO SIMULATION RESULTS OF EXECUTING EACH 100 GENERATED PATHS ARE LISTED WITH THE MEAN AND STANDARD DEVIATION VALUES

Uncertainty considered	$s_{x,thre}$	Max. slip	Energy (kJ)	Success rate 1	Success rate 2	Planning run time (s)
Yes (Proposed; $\delta_{safe} = 0.95$)	0.8	0.651±0.025	23.43±4.60	0.976±0.018	1.000±0.000	305.3±131.7
No	0.8	0.885±0.031	18.81±4.51	0.132±0.087	0.902±0.073	4.4±1.4
No	0.5	0.617±0.037	32.41±5.85	0.038±0.081	1.000±0.000	74.8±27.2

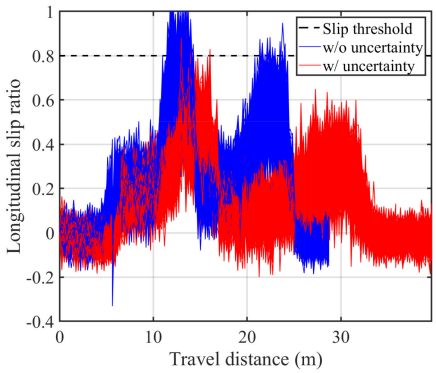


Fig. 7. Sample slip profiles in the 500 Monte Carlo path execution simulations. Without the slip uncertainty consideration during planning (blue), the rover is highly likely to experience a slip higher than the threshold (0.8) during the path execution. With the proposed planning approach (red), the possibility of excessive slip can be bounded.

could be found as depicted in green in Fig. 6(a). These paths never experienced immobilization (success rate 2 = 1.0) in the Monte Carlo simulations, although the slippage exceeded the imposed threshold (mean success rate 1 = 0.038) as listed in Table I. From this observation, incorporating the probabilistic boundary in the constraint Eq.(4) works similar to imposing a conservative constraint in the deterministic approach Eq.(3). The advantage of the proposed robust approach over that in practice is that the amount of the constraint violation, or safety, is probabilistically guaranteed. Additionally, this probabilistic safety level can be specified flexibly depending on user preference and mission requirements, as demonstrated in the next simulation set.

B. Simulation 2: Parameter Studies

1) Setting and Procedures: In the second simulation set, the influence of the two user-specified safety parameters δ_{safe} and $s_{x,thre}$ were evaluated. For the evaluation of δ_{safe}, it was set to 0.99, 0.90, 0.80, or 0.50, while $s_{x,thre}$ was fixed to 0.60. In the evaluation of $s_{x,thre}$, this parameter was set to 0.4, 0.6, or 0.8, while $\delta_{safe} = 0.8$.

The start and goal locations were set to (1.0,1.0) and (17.0,1.0) of the plateau terrain, respectively, with the initial rover heading set toward the goal. Any rover heading at the goal location was accepted. Similarly to the first simulation set, 100 paths were generated for each condition, and then 500 Monte Carlo simulations of path execution were run for each path.

2) Results: The sample paths generated with the varied δ_{safe} are depicted in Fig. 8, and the statistical results of 100 paths are listed in Table II. As seen in the figure, the path generated with a higher δ_{safe} tends to take a longer, shallower route, and the corresponding energy consumption increases. This is because the tighter probability bound of the slip constraint makes the path search to prefer shallower diagonal slope-ascent motion.

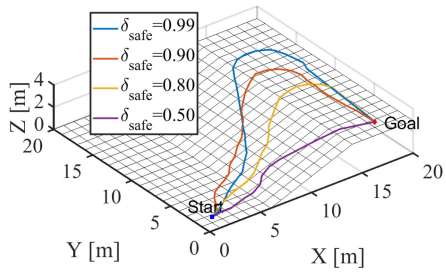


Fig. 8. Sample paths generated with varied probability bound δ_{safe}. With a tighter bound, the resultant path tends to take longer and shallower diagonal slope ascent.

TABLE II
RESULTS OF PATH PLANNING WITH VARIED PROBABILITY BOUND δ_{safe} WITH A FIXED $s_{x,thre} = 0.6$

δ_{safe}	Max. slip	Energy (kJ)	Success rate 1	Run time (s)
0.99	0.442±0.012	13.82±1.15	0.992±0.009	33.07±10.24
0.90	0.510±0.014	12.56±1.27	0.934±0.023	17.37±2.35
0.80	0.542±0.013	11.20±1.05	0.843±0.042	14.57±2.30
0.50	0.584±0.010	8.89±0.28	0.639±0.061	10.45±1.20

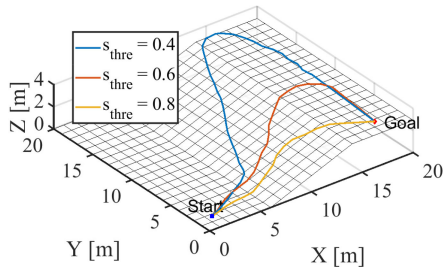


Fig. 9. Sample paths generated with varied slip threshold $s_{x,thre}$. The resultant path takes a shallower route with a lower, more severe slip threshold.

TABLE III
RESULTS OF PATH PLANNING WITH VARIED SLIP THRESHOLD $s_{x,thre}$ WITH A FIXED $\delta_{safe} = 0.8$

$s_{x,thre}$	Max. slip	Energy (kJ)	Success rate 1	Run time (s)
0.40	0.349±0.006	15.65±0.32	0.848±0.028	210.42±82.18
0.60	0.542±0.013	11.20±1.05	0.843±0.042	14.57±2.30
0.80	0.584±0.009	8.95±0.29	0.999±0.001	9.97±1.03

The success rate increases accordingly. Increasing the safety confidence δ_{safe} increases the run-time of path planning.

An example of paths computed with a different slip threshold $s_{x,thre}$ is plotted in Fig. 9. Similar to tightening the probability bound, a lower slip threshold resulted in longer routes with lower longitudinal slip. Table III lists the statistical quantities of the 100 paths. Note that with the slip threshold of $s_{x,thre} = 0.4$, the planner could find 34 paths in the 100 trials within 20000

iterations. The values on the table corresponding to this condition were derived from those successful trials. The planner never failed to find solutions with the other conditions.

In summary, the above results show that the safety level of the resultant paths can be determined by two parameters δ_{safe} and $s_{x,thre}$ in the chance constraint expression. The proposed algorithm provides the flexibility of how much safety against immobilization needs to be guaranteed.

When selecting the safety parameters, it is better to start with conservative values (e.g., $\delta_{safe} > 0.95$ and $s_{x,thre} < 0.5$). If no path is found with the constraint, users might want to repeat easing the regulation values and planning until an acceptable solution is found or change the goal location.

V. Conclusion

This letter proposes a robust path planning algorithm for finding a safe and efficient path on sandy slopes where high slippage is inevitable and there are uncertainties in slip prediction. The proposed algorithm searches for an energy-efficient path with its safety against extremely high slip or immobilization being probabilistically guaranteed based on a chance-constrained planning approach. The level of safety can be determined by the user-specified parameters. The simulation results demonstrated the effectiveness and flexibility of the proposed algorithm.

A critical issue to be addressed in future work is how to efficiently re-plan a path when the terrain maps and traversability models are updated while driving. Another possible direction of future research includes incorporating uncertainties in the rover position and orientation and considering multiple risks (e.g., collision and tip-over) besides the immobilization risk. The authors are planning to improve the proposed algorithm, implement it on a physical rover, and conduct experiments in a planetary-analog test field.

References

[1] M. Ono *et al.*, "Mars 2020 site-specific mission performance analysis: Part 2. Surface traversability," in *Proc. Amer. Inst. Aeronaut. Astronaut. Space*, 2018, pp. 1–14.

[2] M. Maimone *et al.*, "Two years of visual odometry on the mars exploration rovers," *J. Field Robot.*, vol. 24, no. 3, pp. 169–186, 2007.

[3] R. E. Arvidson *et al.*, "Mars science laboratory Curiosity rover megaripple crossings up to sol 710 in gale crater," *J. Field Robot.*, vol. 34, no. 3, pp. 495–518, 2017.

[4] H. Inotsume *et al.*, "Finding routes for efficient and successful slope ascent for exploration rovers," in *Proc. Int. Symp. Artif. Intell., Robot. Automat. Space*, 2016, pp. 1–8.

[5] S. Singh *et al.*, "Recent progress in local and global traversability for planetary rovers," in *Proc. Millennium IEEE Int. Conf. Robot. Autom. Symposia*, 2000, pp. 1194–1200.

[6] A. Kelly *et al.*, "Toward reliable off road autonomous vehicles operating in challenging environments," *Int. J. Robot. Res.*, vol. 25, no. 5–6, pp. 449–483, 2006.

[7] D. Helmick *et al.*, "Terrain adaptive navigation for planetary rovers," *J. Field Robot.*, vol. 26, no. 4, pp. 391–41, 2009.

[8] T. M. Howard and A. Kelly, "Optimal rough terrain trajectory generation for wheeled mobile robots," *Int. J. Robot. Res.*, vol. 26, no. 2, pp. 141–166, 2007.

[9] G. Ishigami, K. Nagatani, and K. Yoshida, "Path planning and evaluation for planetary rovers based on dynamic mobility index," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2011, pp. 601–606.

[10] M. Erdmann, "On motion planning with uncertainty," AI Lab, MIT, Cambridge, MA, USA, Tech. Rep. 810, 1984.

[11] N. Roy, W. Burgard, D. Fox, and S. Thrun, "Coastal navigation-mobile robot navigation with uncertainty in dynamic environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1999, vol. 1, pp. 35–40.

[12] J. P. Gonzalez and A. Stentz, "Planning with uncertainty in position: An optimal and efficient planner," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2005, pp. 2435–2442.

[13] G. Kewlani, G. Ishigami, and K. Iagnemma, "Stochastic mobility-based path planning in uncertain environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2009, pp. 1183–1189.

[14] N. A. Melchior and R. Simmons, "Particle RRT for path planning with uncertainty," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2007, pp. 1617–1624.

[15] L. Blackmore, H. Li, and B. Williams, "A probabilistic approach to optimal robust path planning with obstacles," in *Proc. IEEE Amer. Control Conf.*, 2006, pp. 2831–2837.

[16] B. D. Luders *et al.*, "Robust sampling-based motion planning with asymptotic optimality guarantees," in *Proc. Amer. Inst. Aeronaut. Astronaut. GNC*, 2013, pp. 1–25.

[17] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 723–730.

[18] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, 2011.

[19] S. U. Lee, R. Gonzalez, and K. Iagnemma, "Robust sampling-based motion planning for autonomous tracked vehicles in deformable high slip terrain," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 2569–2574.

[20] C. Urmson and R. Simmons, "Approaches for heuristically biasing RRT growth," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2003, vol. 2, pp. 1178–1183.

[21] J. Y. Wong, *Theory of Ground Vehicles.* Hoboken, NJ, USA: Wiley, 1978.

[22] A. Angelova *et al.*, "Learning and prediction of slip from visual information," *J. Field Robot.*, vol. 24, no. 3, pp. 205–231, 2007.

[23] G. Ishigami, G. Kewlani and K. Iagnemma, "Predictable mobility," *IEEE Robot. Autom. Mag.*, vol. 16, no. 4, pp. 61–70, Dec. 2009.

[24] C. Cunningham, M. Ono, I. Nesnas, J. Yen, and W. L. Whittaker, "Locally-adaptive slip prediction for planetary rovers using Gaussian processes," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 5487–5494.

[25] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.

[26] G. Sakayori and G. Ishigami, "Energy efficient slope traversability planning for mobile robot in loose soil," in *Proc. IEEE Int. Conf. Mechatronics*, 2017, pp. 99–104.